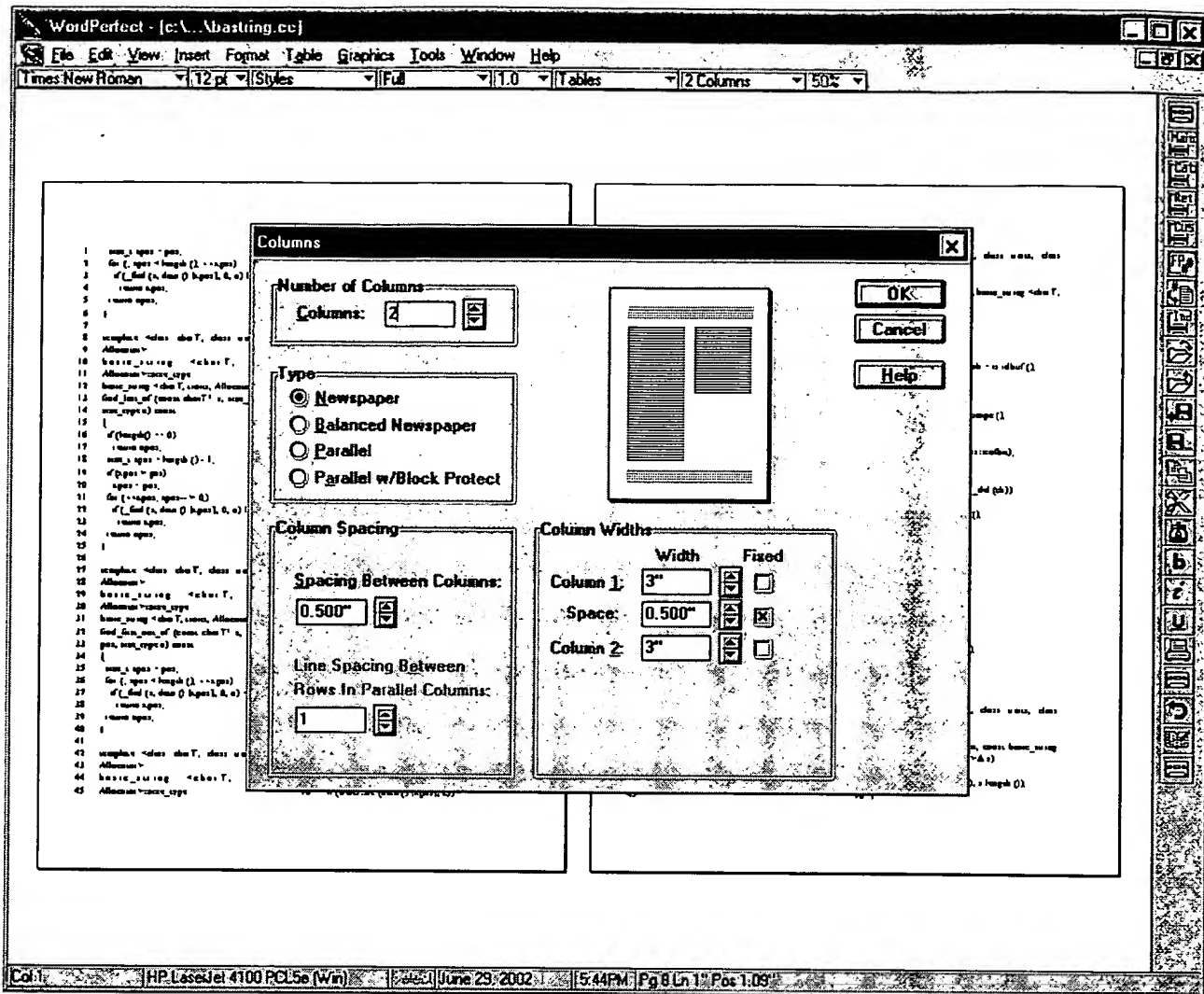
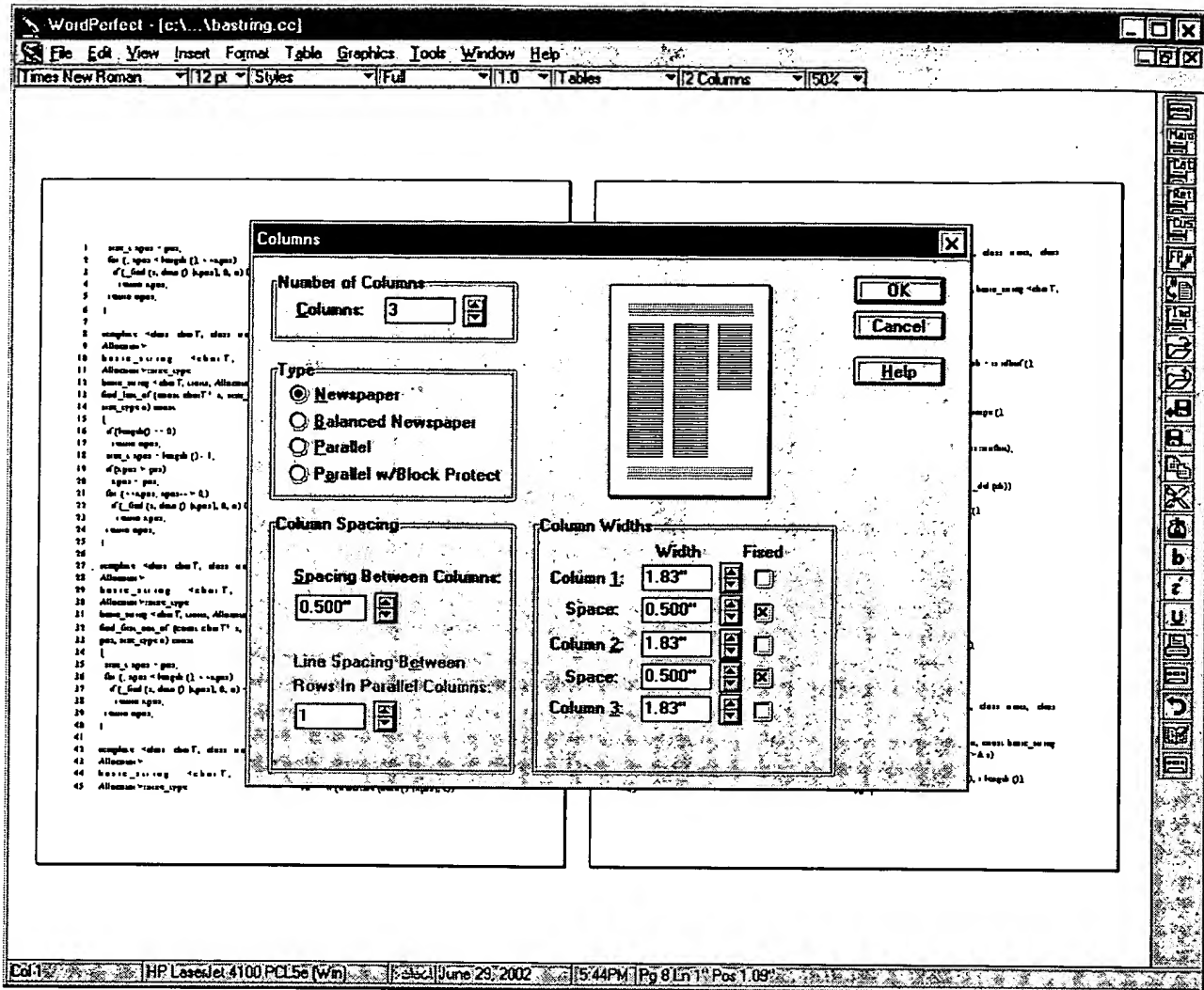
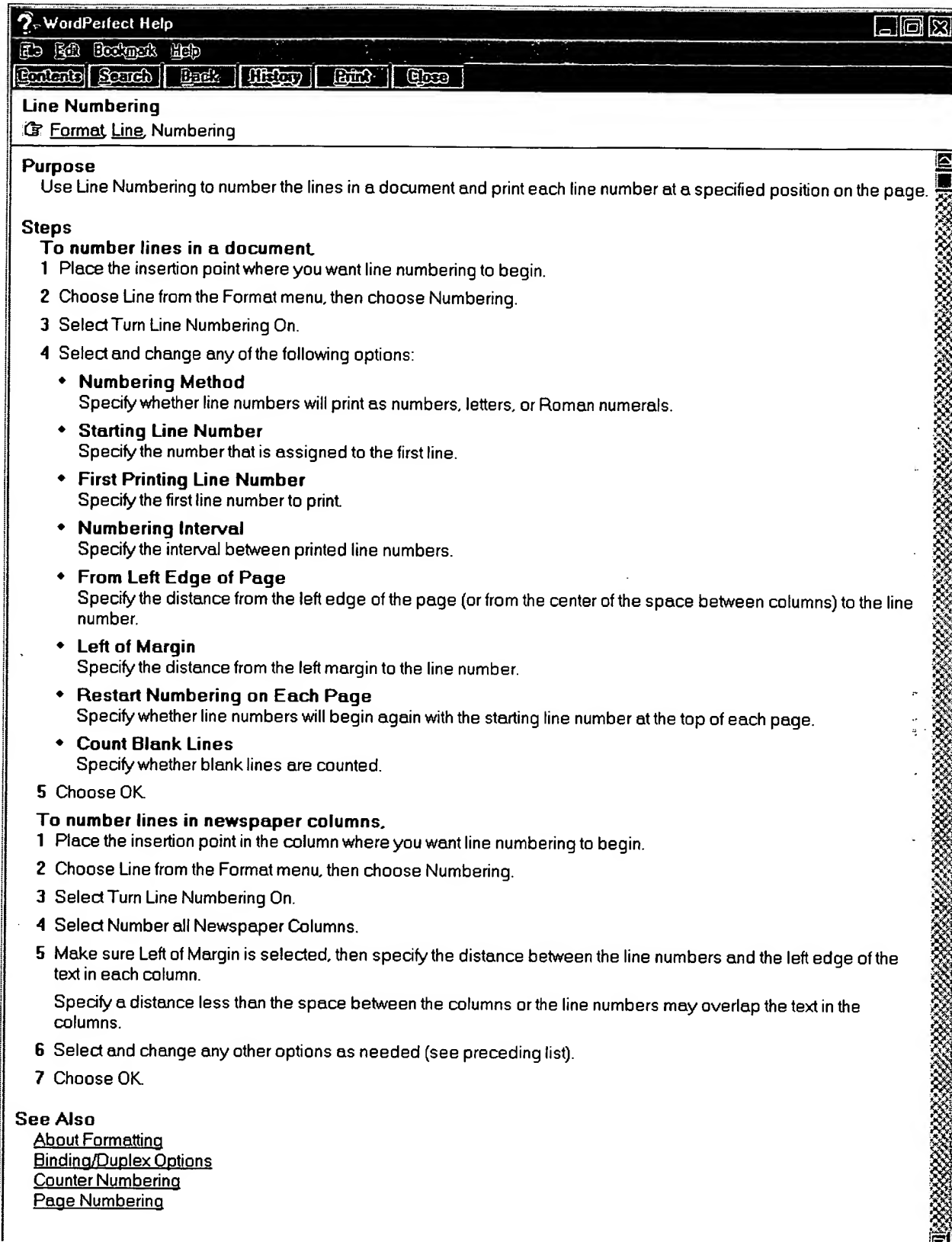


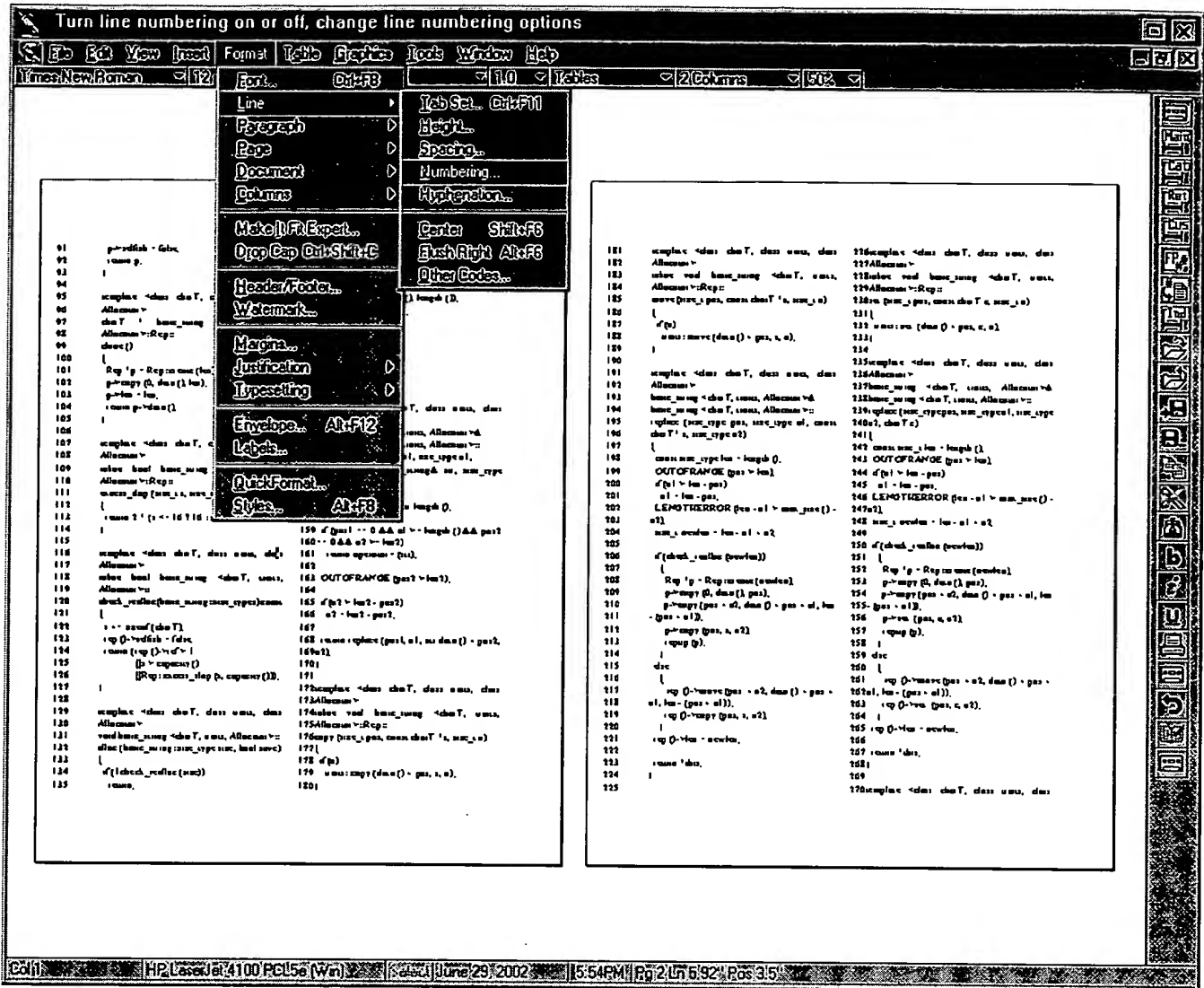
Columns	Define...
Make/Unfit Expr...	Off
Drop/Exp. Crit/Stat-C	Column Break
Header/Footer...	Border/L...
Watermark...	
Margins...	(1 + signs)
Justification	(1 sign, 1 d)
Hyphenating	
Envelope... Alt-F12	
Labels...	
QuickFormat...	
Styles... Alt-F8	

[illegible]









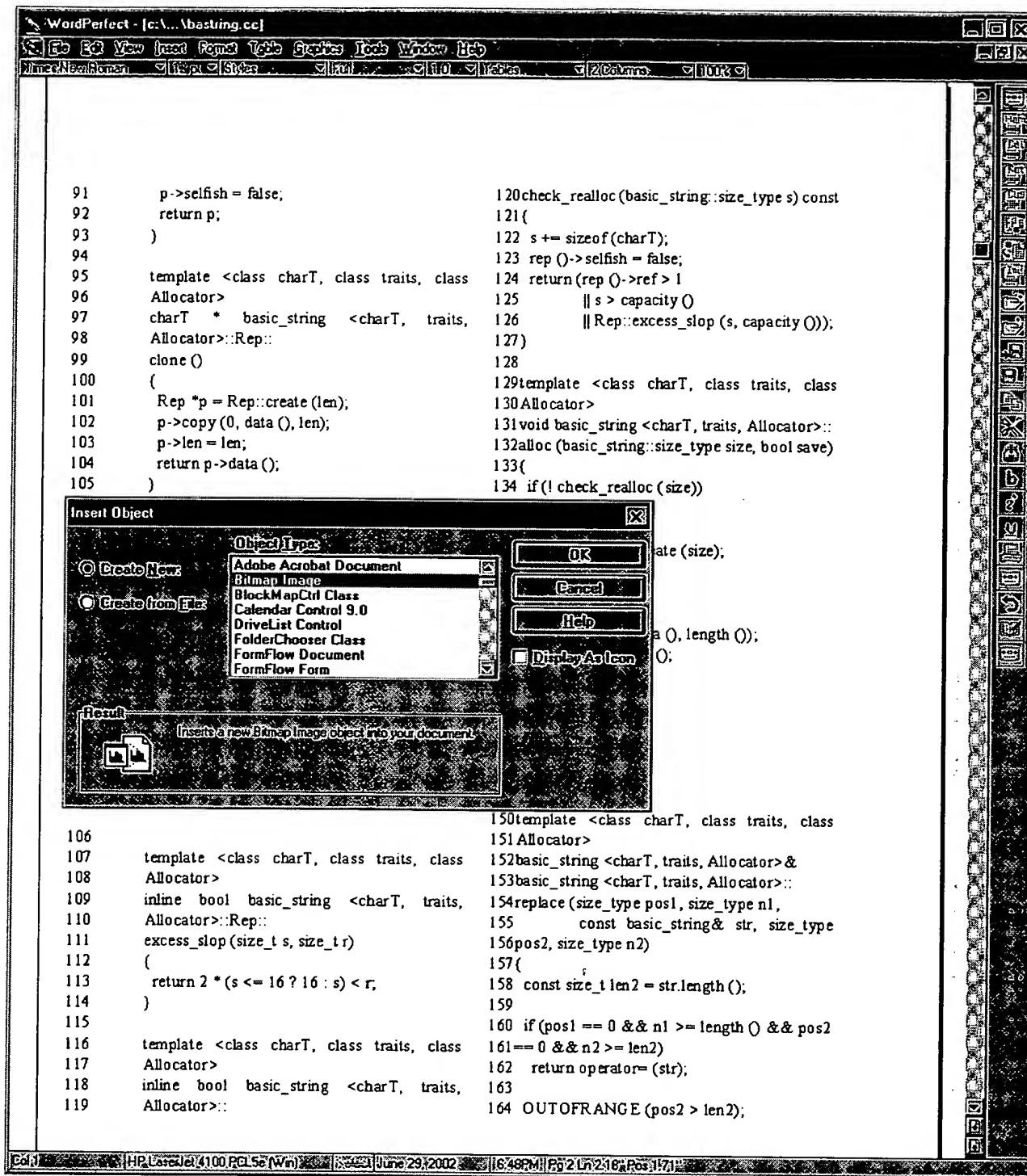
1 // Member template for the "C++" string
2 class.
3 // Copyright (C) 1994 Free Software
4 Foundation
5
6 // This file is part of the GNU ANSI C++
7 Library. This library is free
8 // software; you can redistribute it under
9 // terms of the GNU General Public License as
10 // published by the
11 // Free Software Foundation, either version 2
12 // or (at your option) any later version.
13
14 // This library is distributed in the hope that it
15 // will be useful,
16 // but WITHOUT ANY WARRANTY
17 // without even the implied warranty of
18 // MERCHANTABILITY or FITNESS FOR
19 // A PARTICULAR PURPOSE. See the
20 // GNU General Public License for more
21 // details.
22
23 // You should have received a copy of the
24 // GNU General Public License
25 // along with this library; see the file
26 // COPYING. If not, write to the Free
27 // Software Foundation, 59 Temple Place -
28 // Suite 300, Boston, MA 02111-1307, USA.
29
30
31
32 // As a special exception, if you link this library
33 // with files
34 // compiled with a GNU compiler to produce
35 // an executable, this does not cause
36 // the resulting executable to be covered by the
37 // GNU General Public License.
38 // This exception does not however invalidate
39 // any other reason why
40 // the executable file might be covered by the
41 // GNU General Public License.
42
43 // Written by Jonon Hestell based upon the
44 // specification by T. Koenig
45 // in ANSI C211/64.003E3.

46 #include "C++"
47 template <class charT, class traits, class
48 Allocator>
49 inline void * basic_string<charT, traits,
50 Allocator>::Rep*
51 { return new(Allocator)(size_t, traits);
52 }
53 template <class charT, class traits, class
54 Allocator>
55 {
56 template <class charT, class traits, class
57 Allocator>
58 inline void * basic_string<charT, traits,
59 Allocator>::Rep*
60 { return new(Allocator)(size_t, traits);
61 }
62 {
63 Allocator::Allocator(ptr, sizeof(Rep) *
64 sizeof(charT));
65 }
66 {
67 }
68
69 template <class charT, class traits, class
70 Allocator>
71 inline size_t basic_string<charT, traits,
72 Allocator>::Rep*
73 { return new(Allocator)(size_t, traits);
74 }
75 {
76 size_t i = 1;
77 while (i < size_t) i = 2 * i;
78 return i;
79 }
80 template <class charT, class traits, class
81 Allocator>
82 inline basic_string<charT, traits,
83 Allocator>::Rep*
84 { return new(Allocator)(size_t, traits);
85 }
86 {
87 size_t i = 1;
88 while (i < size_t) i = 2 * i;
89 return i;
90 }
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135

136 Rep* p = Rep::create(size);
137 return p;
138 }
139 {
140 p->copy(0, data(0), length(0));
141 p->len = length(0);
142 }
143 {
144 p->len = 0;
145 }
146 {
147 }
148 }
149 template <class charT, class traits, class
150 Allocator>
151 basic_string<charT, traits, Allocator> &
152 basic_string<charT, traits, Allocator>::
153 replace(size_type pos1, size_type n1,
154 const basic_string& str, size_type
155 pos2, size_type n2)
156 {
157 const size_t len2 = str.length(0);
158 if (pos1 == 0 && n1 > length(0) && pos2
159 == 0 && n2 > len2)
160 return operator=(str);
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180

```
91 p->init = false;
92 return p;
93 }
94
95 template <class charT, class traits, class
96 Allocator>
97 charT * basic_string<charT, traits,
98 Allocator>::Rep::
99 clone()
100 {
101     Rep *p = Rep::create(len);
102     p->copy(0, data(), len);
103     p->len = len;
104     return p->data();
105 }
106
107 template <class charT, class traits, class
108 Allocator>
109 inline bool basic_string<charT, traits,
110 Allocator>::Rep::
111 access_log(int i, int n)
112 {
113     return 2 * (i <= 16 ? i) < n;
114 }
115
116 template <class charT, class traits, class
117 Allocator>
118 inline bool basic_string<charT, traits,
119 Allocator>::
120 check_locale(basic_string<int_type>::const
121 iterator, const charT*, class traits, class
122 Allocator>::Rep::
123 iterator, const charT*, traits, Allocator>::
124 iterator, const charT*, traits, Allocator>::
125 iterator, const charT*, traits, Allocator>::
126 iterator, const charT*, traits, Allocator>::
127 iterator, const charT*, traits, Allocator>::
128 iterator, const charT*, traits, Allocator>::
129 iterator, const charT*, traits, Allocator>::
130 iterator, const charT*, traits, Allocator>::
131 iterator, const charT*, traits, Allocator>::
132 iterator, const charT*, traits, Allocator>::
133 iterator, const charT*, traits, Allocator>::
134 iterator, const charT*, traits, Allocator>::
135 iterator, const charT*, traits, Allocator>::
```

```
136 Rep *p = Rep::create(len);
137 Allocator>
138 {
139     p->copy(0, data(), length());
140     p->len = length();
141     return p;
142 }
143
144 template <class charT, class traits, class
145 Allocator>
146 inline bool basic_string<charT, traits,
147 Allocator>::
148 access_log(int i, int n)
149 {
150     return 2 * (i <= 16 ? i) < n;
151 }
152
153 template <class charT, class traits, class
154 Allocator>
155 inline bool basic_string<charT, traits,
156 Allocator>::
157 access_log(int i, int n)
158 {
159     return 2 * (i <= 16 ? i) < n;
160 }
161
162 template <class charT, class traits, class
163 Allocator>
164 inline bool basic_string<charT, traits,
165 Allocator>::
166 access_log(int i, int n)
167 {
168     return 2 * (i <= 16 ? i) < n;
169 }
170
171 template <class charT, class traits, class
172 Allocator>
173 inline bool basic_string<charT, traits,
174 Allocator>::
175 access_log(int i, int n)
176 {
177     return 2 * (i <= 16 ? i) < n;
178 }
179
180 template <class charT, class traits, class
181 Allocator>
182 inline bool basic_string<charT, traits,
183 Allocator>::
184 access_log(int i, int n)
185 {
186     return 2 * (i <= 16 ? i) < n;
187 }
188
189 template <class charT, class traits, class
190 Allocator>
191 inline bool basic_string<charT, traits,
192 Allocator>::
193 access_log(int i, int n)
194 {
195     return 2 * (i <= 16 ? i) < n;
196 }
197
198 template <class charT, class traits, class
199 Allocator>
200 inline bool basic_string<charT, traits,
201 Allocator>::
202 access_log(int i, int n)
203 {
204     return 2 * (i <= 16 ? i) < n;
205 }
206
207 template <class charT, class traits, class
208 Allocator>
209 inline bool basic_string<charT, traits,
210 Allocator>::
211 access_log(int i, int n)
212 {
213     return 2 * (i <= 16 ? i) < n;
214 }
215
216 template <class charT, class traits, class
217 Allocator>
218 inline bool basic_string<charT, traits,
219 Allocator>::
220 access_log(int i, int n)
221 {
222     return 2 * (i <= 16 ? i) < n;
223 }
224
225 template <class charT, class traits, class
```



WordPerfect - [c:\...bastring.cc]

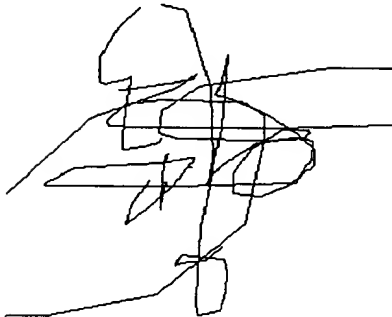
File Edit View Insert Format Table Graphics Tools Window Help

Times New Roman 12pt 1/2" 1/2" 100%

```

91     p->selfish = false;
92     return p;
93 }
94
95 template <class charT, class traits, class
96 Allocator>
97 charT * basic_string <charT, traits,
98 Allocator>::Rep::
99 clone ()
100 {
101     Rep *p = Rep::create (len);
102     p->copy (0, data (), len);
103     p->len = len;
104     return p->data ();
105 }

```



```

106
107 template <class charT, class traits, class
108 Allocator>
109 inline bool basic_string <charT, traits,
110 Allocator>::Rep::
111 excess_slop (size_t s, size_t r)
112 {
113     return 2 * (s <= 16 ? 16 : s) < r;
114 }
115
116 template <class charT, class traits, class
117 Allocator>
118 inline bool basic_string <charT, traits,
119 Allocator>::

```

```

120 check_realloc (basic_string::size_type s) const
121 {
122     s += sizeof (charT);
123     rep ()->selfish = false;
124     return (rep ()->ref > 1
125         || s > capacity ()
126         || Rep::excess_slop (s, capacity ()));
127 }
128
129 template <class charT, class traits, class
130 Allocator>
131 void basic_string <charT, traits, Allocator>::
132 alloc (basic_string::size_type size, bool save)
133 {
134     if (! check_realloc (size))
135         return;
136
137     Rep *p = Rep::create (size);
138
139     if (save)
140     {
141         p->copy (0, data (), length ());
142         p->len = length ();
143     }
144     else
145         p->len = 0;
146
147     repup (p);
148 }
149
150 template <class charT, class traits, class
151 Allocator>
152 basic_string <charT, traits, Allocator> &
153 basic_string <charT, traits, Allocator>::
154 replace (size_type pos1, size_type n1,
155         const basic_string& str, size_type
156 pos2, size_type n2)
157 {
158     const size_t len2 = str.length ();
159
160     if (pos1 == 0 && n1 >= length () && pos2
161         == 0 && n2 >= len2)
162         return operator= (str);
163
164     OUTOFRANGE (pos2 > len2);

```

Col 2 HP LaserJet 4100 PCL5e (Win) June 29, 2002 6:47 PM Pg 2 Ln 493 Pos 475

WordPerfect - [c:\test\wpd]

File Edit View Insert Format Table Graphics Tools Window Help

Times New Roman 11pt Left 12.0 Columns 100%

Application/Control Number: 11/111,111 Page 2

Art Unit: 2176

1 Conclusion 25 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

2 4. Prior art made of record and not relied 26 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

3 upon is considered pertinent to disclosure. 27 XXXXXXXXXXXXXXXX to the

4 Sotomayor U.S. Patent No. 28 Group receptionist whose telephone number is

5 5,708,825 issued January

6 1998

7

8 5. Any inquiry concerning should be

9 communication or earlier c

10 the examiner

11 should be directed to William L. Bashore, Jr. and

12 telephone number is (703) 746-7238 0231

13 examiner can

14 normally be reached on Monday, December 23, 2002, from 11:30 AM to 8:00 PM

15 from 11:30 AM to 8:00 PM

16 If attempts to reach the receptionist are unsuccessful, the examiner's supervisor, Heather Herndon, can be reached on (703) 308-5186.

17 telephone are unsuccessful, the examiner's supervisor, Heather Herndon, can be reached on (703) 308-5186.

18 supervisor,

19 Heather Herndon, can be reached on (703) 308-5186.

20 5186.

21 Any inquiry of a general nature or relating to the status of this application should be directed to William L. Bashore, Jr. at (703) 746-7238.

22 to the status of this application should be directed

23 XXXXXXXXXXXX

24 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

45 or: (703) 746-7238 (for after-final communications)

46

47 communications)

48

49 Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA, Fourth Floor (Receptionist).

50 brought to Crystal Park II, 2121 Crystal Drive,

51 Drive,

52 Arlington, VA, Fourth Floor

53 (Receptionist).

54

55 William L. Bashore

56

57 10/14/2002

HP LaserJet 4100 PCL5e (Win) [362] December 23, 2002 12:56 PM Page 2 of 2

09/371,716

(15)

Art Unit: 2176

24 Group receptionist whose telephone number is

25 (703) 305-3900.

26

27 6. Any response to this action should be

28 mailed to:

29 Commissioner of Patents and

30 Trademarks

31 Washington, D.C. 20231

32

33 or faxed to:

34 (703) 746-7239 (for formal

35 communications intended for entry)

36 or:

37 (703) 746-7240 (for informal or

38 draft communications, please label

39 "PROPOSED" or

40 "DRAFT")

41 or:

42 (703) 746-7238 (for after-final

43 communications)

44

45 **Hand-delivered responses should be**

46 brought to Crystal Park II, 2121 Crystal

47 Drive,

48 Arlington, VA, Fourth Floor

49 (Receptionist).


50

51 William L. Bashore

52

53 10/14/2002

to the

 (16)

Application/Control Number: 11/111,111

Page 2

Art Unit: 2176

1 *Conclusion* 25 Group receptionist whose telephone number is
2 4. Prior art made of record and not relied 26 (703) 305-3900.
3 upon is considered pertinent to disclosure. 27
4 Sotomayor U.S. Patent No. 28 6. Any response to this action should be
5 5,708,825 issued January 29 mailed to:
6 1998
7
8 5. Any inquiry concerning this 30 Commissioner of Patents and
9 communication or earlier communications from 31 Trademarks
10 the examiner 32 Washington, D.C. 20231
11 should be directed to William Bashore whose 33
12 telephone number is (703) 308-5807. The 34 or faxed to:
13 examiner can 35 (703) 746-7239 (for formal
14 normally be reached on Monday through Friday 36 communications intended for entry)
15 from 11:30 AM to 8:00 PM EST. 37 or:
16 If attempts to reach the examiner by 38 (703) 746-7240 (for informal or
17 telephone are unsuccessful, the examiner's 39 draft communications, please label
18 supervisor, 40 "PROPOSED" or
19 Heather Herndon, can be reached on (703) 308- 41 "DRAFT")
20 5186. 42 or:
21 Any inquiry of a general nature or relating 43 (703) 746-7238 (for after-final
22 to the status of this application should be directed 44 communications)
23 XXXXXXXXX 45
24 XXXXXXXXXXXXXXXXXXXXXXXX to the 46 Hand-delivered responses should be
47 brought to Crystal Park II, 2121 Crystal
48 Drive,
49 Arlington, VA, Fourth Floor
50 (Receptionist).
51
52 William L. Bashore
53
54 10/14/2002

(17)

WordPerfect - [c:\test.wpd]

File Edit View Insert Format Table Graphics Tools Window Help

Times New Roman 11 pt Styles Left 2.0 Tables 2 Columns 100%

Application/Control Number: 11/111,111 Page 2

Art Unit: 2176

1 *Conclusion* 25 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

2 4. Prior art made of record and not relied 26 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

3 upon is considered pertinent to disclosure. 27 XXXXXXXXXXXXXXXto the

4 Sotomayer U.S. Patent No. 28 Group receptionist whose telephone number is

5 5,708,825 issued January 29 (703) 305-3900.

6 1998

7

8 5. Any inquiry concerning this 30

9 communication or earlier communications from 31 6. Any response to this action should be

10 the examiner 32 mailed to:

11 should be directed to William Bashore whose 33 Commissioner of Patents and

12 telephone number is (703) 308-5807. The 34 Trademarks

13 examiner can 35 Washington, D.C. 20231

14 normally be reached on Monday through Friday 36

15 from 11:30 AM to 8:00 PM EST. 37 or faxed to:

16 If attempts to reach the examiner by 38 (703) 746-7239 (for formal

17 telephone are unsuccessful, the examiner's 39 communications intended for entry)

18 supervisor, 40 or:

19 Heather Herndon, can be reached on (703) 308- 41 (703) 746-7240 (for informal or

20 5186. 42 draft communications, please label

21 Any inquiry of a general nature or relating 43 "PROPOSED" or

22 to the status of this application should be directed 44 "DRAFT")

23 XXXXXXXXXX 45 or:

24 XXXXXXXXXXXXXXXXXXXXXXXXXXXX 46 (703) 746-7238 (for after-final

47 communications)

48

49 Hand-delivered responses should be

50 brought to Crystal Park II, 2121 Crystal

51 Drive,

52 Arlington, VA, Fourth Floor

53 (Receptionist).

54

55 William L. Bashore

56

57 10/14/2002

CM2 HP LaserJet 4100 PCL5e (Win) 2002/12/23/2:55PM P52 L7279 P65599

(18)

ProQuest®[Return to NPL Web Page](#)[Text Version](#)[?Help](#)Searching collections: **All Collections****Search Results****Save Link** Saves this search as a Durable Link under "Results-Marked List"**100 articles matched your search. At least 11 articles are full text.**

- ☐ 1. Code season; *Christopher Lindquist*; **Computerworld**, Framingham; Dec 7, 1998; Vol. 32, Iss. 49; pg. 65, 2 pgs
- ☐ 2. Change of address; *Karen L Miller*; **Home Office Computing**, Boulder; Dec 1998; Vol. 16, Iss. 12; pg. 102, 2 pgs
- ☐ 3. Gambling with Visual J++ 6.0; *Al Williams*; **Web Techniques**, San Francisco; Dec 1998; Vol. 3, Iss. 12; pg. 32, 1 pgs
- ☐ 4. IE 5.0 Simplifies Searches; *Yael Li-Ron*; **PC World**; **PC World Online**, San Francisco; Nov 3, 1998; pg. 1
- ☐ 5. JavaScript and ModelServer Publisher; *Cordell Brown*; **MicroStation Manager**, Madison; Oct 1998; Vol. 8, Iss. 10; pg. S1, 6 pgs
- ☐ 6. Netscape Announces Netscape Application Builder 3.0 To Speed Development Of Applications For Netscape Application Server; **PR Newswire**, New York; Sep 22, 1998; pg. 1
- ☐ 7. Meridian Partners Limited Produces Web Components for Microsoft Visual Studio 6.0; **PR Newswire**, New York; Sep 8, 1998; pg. 1
- ☐ 8. Macromedia Freehand and Fireworks Win eMediaweekly Digital Media Awards; **PR Newswire**, New York; Sep 2, 1998; pg. 1
- ☐ 9. Macromedia Announces Dreamweaver and Fireworks 'Web Essentials' Promotion; **PR Newswire**, New York; Sep 1, 1998; pg. 1
- ☐ 10. Godzilla vs. your system; *Eric Foster-Johnson*; **UNIX Review's Performance Computing**, San Francisco; Sep 1998; Vol. 16, Iss. 10; pg. 63, 4 pgs

[Explain Icons](#)**Next**

11-20

Enter a word, words or specific phrase.

SearchDate range: Publication type: Search in:

- ☒ Show results with full text availability only
- ☐ Show peer reviewed publications
- ☒ Show total number of articles

09/371,716

ProQuest®[Return to NPL Web Page](#)[Text Version](#)[? Help](#)Searching collections: **All Collections****Search Results****Save Link** Saves this search as a Durable Link under "Results-Marked List"**100 articles matched your search. At least 21 articles are full text.**

- ☐ 11. Macromedia and Allaire Announce New Cold Fusion(R) Objects for Dreamweaver(TM); PR Newswire, New York; Aug 18, 1998; pg. 1
- ☐ 12. Ways of working with browsers; Computimes Malaysia, New York; Aug 17, 1998; pg. 1
- ☐ 13. Ways of working with browsers; [Computimes, 2* Edition]; Maria O'Daniel; New Straits Times, Kuala Lumpur; Aug 17, 1998; pg. 39
- ☐ 14. Structure provides Web's glue; [SOUTH PINELLAS Edition]; JULES ALLEN; St. Petersburg Times, St. Petersburg, Fla.; Aug 17, 1998; pg. 13
- ☐ 15. Introducing Visual InterDev 6.0; Anonymous; Active Server Developer's Journal, Louisville; Aug 1998; Vol. 2, Iss. 8; pg. 1, 6 pgs
- ☐ 16. Web browsers: Microsoft Corp.; Anonymous; UNIX Review's Performance Computing, San Francisco; Aug 1998; Vol. 16, Iss. 9; pg. 78, 1 pgs
- ☐ 17. Choosing a Macintosh Web authoring tool: It's not that easy these days; Shelly Brisbin; Web Techniques, San Francisco; Aug 1998; Vol. 3, Iss. 8; pg. 59, 4 pgs
- ☐ 18. SunWorld Magazine Reports Results of Reader Survey On Open Source In the Enterprise; PR Newswire, New York; Jul 29, 1998; pg. 1
- ☐ 19. Controlling your Web site; Brooks Talley; InfoWorld, San Mateo; Jul 27, 1998; Vol. 20, Iss. 30; pg. 66, 4 pgs
- ☐ 20. Oracle(R) Tools Offer Immediate and Integrated Support for Oracle8(TM); PR Newswire, New York; Jul 1, 1998; pg. 1

[Explain Icons](#)

First **Previous** **Next**

1-10 21-30

Enter a word, words or specific phrase.

 Date range: Publication type: Search in:

- ☒ Show results with full text availability only
- ☐ Show peer reviewed publications
- ☒ Show total number of articles

[Return to NPL Web Page](#)[Text Version](#)[?Help](#)Searching collections: **All Collections****Search Results**[Save Link](#) Saves this search as a Durable Link under "Results-Marked List"**100 articles matched your search. At least 31 articles are full text.**

- ☐ 21. [Dreamweaver: A new state of consciousness?](#); *Sue Wilcox*; **Web Techniques**, San Francisco; Jul 1998; Vol. 3, Iss. 7; pg. 59, 4 pgs
- ☐ 22. [Visual J++ 6.0: Doing Java the Windows way](#); *Eric Binary Anderson*; **ENT**, Fort Washington; Jun 24, 1998; Vol. 3, Iss. 11; pg. 52, 1 pgs
- ☐ 23. [GOLIVE SYSTEMS: GoLive Systems tackles the consumer market](#); **M2 Presswire**, Coventry; Jun 4, 1998; pg. 1
- ☐ 24. [Macromedia's Dreamweaver 1.0](#); *Stephen Ellerin*; **E Media Professional**, Wilton; Jun 1998; Vol. 11, Iss. 6; pg. 73, 2 pgs
- ☐ 25. [GoLive Systems Tackles the Consumer Market](#); **PR Newswire**, New York; May 29, 1998; pg. 1
- ☐ 26. [Sapphire/Web: A gem of an integration solution](#); *Roseann McGrath Brooks*; **ENT**, Fort Washington; May 20, 1998; Vol. 3, Iss. 9; pg. 69, 1 pgs
- ☐ 27. [Bluestone Software Sapphire/Web Release 5](#); *Anonymous*; **Network Computing**, Manhasset; May 15, 1998; Vol. 9, Iss. 9; pg. 192, 1 pgs
- ☐ 28. [MACROMEDIA: Macromedia introduces Dreamweaver 1.2](#); **M2 Presswire**, Coventry; Apr 22, 1998; pg. 1
- ☐ 29. [Macromedia Introduces Dreamweaver\(TM\) 1.2](#); **PR Newswire**, New York; Apr 21, 1998; pg. 1
- ☐ 30. [Painless Web Pages](#); *Michael Goodwin and Sam Hammond*; **PC World Online**, San Francisco; Apr 19, 1998; pg. 1

[Explain Icons](#)

11-20 31-40

Enter a word, words or specific phrase.

 [Search](#)Date range: ▼Publication type: ▼Search in: ▼

- ☒ Show results with full text availability only
- ☐ Show peer reviewed publications
- ☒ Show total number of articles